ER0195
Errata
SmartFusion2 M2S050 (T,TS)

Microsemi
a Microchip company

Microsemi Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at www.microsemi.com.

Contents

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 1.6

Section Concurrent access of Cortex-M3 I and D-buses is not allowed, page 7 is updated in revision 1.6 of this document.

1.2 Revision 1.5

Errata item 23, User eNVM write is incorrectly allowed for the protected 4K Regions, page 9 is added in the revision 1.5 of this document.

1.3 Revision 1.4

Errata item 22, For S (security) grade devices, user must not enable write protection for protected 4 K Regions, also known as Special Sectors in the eNVM, page 8, is added in revision 1.4 of this document.

1.4 Revision 1.3

Table 3, page 10 and Table 4, page 10: Auto Programming and 2 Step IAP use SC_SPI programming interface are updated in revision 1.3 of this document.

1.5 Revision 1.2

The following items are added in revision 1.2 of this document:

- Information about Revision 2 of the M2S050 device
- Errata item 20, Disable Cortex-M3 when programming eNVM only, page 8 and errata item 21, The DDR I/Os in M2S050 (T,TS)-FG896 are non-compliant with the DDR3 standard, page 8 are added
- Table 4, page 10

1.6 Revision 1.1

Updated to remove IAP from item 19, After successful completion of CM3 ISP (without a SYSRESET), LSRAM Read and Write access fails from the fabric path, page 8 in revision 1.1 of this document.

1.7 Revision 1.0

Revision 1.0 was the first publication of this document. All M2S050 (T,TS) device errata are combined in this revision.

2 Errata for SmartFusion2 M2S050 (T,TS) SoC FPGA Commercial and Industrial Devices

2.1 Revisions Released per Device

The following table describes the revisions released per device and the device status.

Table 1 • **Revisions Released per Device**

Silicon Devices	Revisions	Device Status
M2S050 (T,TS)	Commercial/Industrial	Production

2.2 Introduction

This errata sheet contains the specific device errata and the affected SmartFusion2® M2S050 (T,TS) revisions of commercial and industrial devices.

2.3 Summary of SmartFusion2 M2S050 (T,TS) Device Errata

The following table gives the summary of SmartFusion2 M2S050 (T,TS) Device errata items:

Table 2 • **Summary of SmartFusion2 M2S050 (T,TS) Device Errata**

		Silicon Revisions M2S050 (T,TS)		
No.	**Errata**	**0,1**	**2**	**Software Errata**
1.	The MDDR and FDDR AXI interface does not support exclusive access, page 4	X	X	–
2.	Apply DEVRST_N after ISP programming, page 4	X	–	–
3.	AXI wrap transfers with more than 32 bytes in burst mode are not supported for MDDR and FDDR, page 4	X	X	–
4.	The MDDR/FDDR controller must be used with the sequential burst mode with BL = 8 and PHY = 32, or PHY = 16, page 4	X	X	–
5.	Use USBD I/O group for USB-ULPI mode in the M2S050-FG896 devices, page 4	X	X	–
6.	VPP must be set to 2.5 V when programming/writing the eNVM at Industrial temperature range, page 5	X	–	–
7.	Over-voltage support on MSIOs during Flash*Freeze mode, page 5	X	–	–
8.	Verification of the FPGA fabric at junction temperatures higher than 50°C erroneously indicates a failure, page 5	X	–	–
9.	DDR_OUT and I/O-Reg functional Errata due to a software bug, page 5	–	–	X

Table 2 • **Summary of SmartFusion2 M2S050 (T,TS) Device Errata *(continued)***

No.	Errata	Silicon Revisions M2S050 (T,TS) 0,1	2	Software Errata
10.	Dedicated differential I/O driving the reference clock of the CCC may cause a functional failure due to a software bug, page 5	–	–	X
11.	NVM Ready bit in eNVM Status register can generate a false READY signal, page 6	X	–	–
12.	Power-up Digest is not supported, page 6	X	–	–
13.	Programming of the eNVM should only occur as part of a bitstream also containing the FPGA fabric, page 6	–	–	X
14.	Updating eNVM from the MSS or the FPGA fabric requires changes of NV_FREQRNG register, page 6	X	X	–
15.	SYSCTRL_RESET_STATUS macro is not supported, page 6	X	X	–
16.	Zeroization is not supported, page 6	X	X	–
17.	Concurrent access of Cortex-M3 I and D-buses is not allowed, page 7	X	X	–
18.	PCIe Hot Reset support requires a soft reset solution, page 7	X	X	–
19.	After successful completion of CM3 ISP (without a SYSRESET), LSRAM Read and Write access fails from the fabric path, page 8	X	X	–
20.	Disable Cortex-M3 when programming eNVM only, page 8	X	X	–
21.	The DDR I/Os in M2S050 (T,TS)-FG896 are non-compliant with the DDR3 standard, page 8	X	X	–
22.	For S (security) grade devices, user must not enable write protection for protected 4 K Regions, also known as Special Sectors in the eNVM, page 8	X	X	–
23.	User eNVM write is incorrectly allowed for the protected 4K Regions, page 9	X	–	–

Note: "X" indicates that the errata exists for that particular device and revision number.

Note: "–" indicates that the errata does not exist or the feature does not exist for that particular device and revision number.

Note: Occurrence of the software errata can be avoided by using the Libero® SoC v11.4 SP1 or later versions.

Note: Contact *Microsemi Global support* if you have additional questions. To order a specific die, contact your local Microsemi sales office.

3 Errata Descriptions and Solutions

3.1 The MDDR and FDDR AXI interface does not support exclusive access

The MDDR and FDDR AXI interface in the M2S050 device is compliant with AMBA AXI Protocol Specification v1.0, except for the exclusive access functionality. The future version of the errata will have updated information about the exclusive access functionality for the AXI interface in the M2S050 Commercial device.

3.2 Apply DEVRST_N after ISP programming

M2S050 Commercial and Industrial devices support device programming in JTAG, Slave SPI, and ISP programming modes. However, after ISP programming, DEVRST_N needs to be asserted to reset the device or power cycle the device to run the new design.

3.3 AXI wrap transfers with more than 32 bytes in burst mode are not supported for MDDR and FDDR

Do not use wrap transfers with more than 32 bytes.

3.4 The MDDR/FDDR controller must be used with the sequential burst mode with BL = 8 and PHY = 32, or PHY = 16

Although the MDDR and FDDR controllers in the M2S050 devices support various burst modes/ lengths and PHY settings (as specified in the *UG0446: SmartFusion2 and IGLOO2 FPGA High Speed DDR Interfaces User Guide)*, only a subset of these settings are supported.

Recommendation:

Only use sequential burst mode with BL = 8 for PHY16, or PHY32 modes for the MDDR or FDDR.

3.5 Use USBD I/O group for USB-ULPI mode in the M2S050-FG896 devices

For interfacing the USB OTG controller with ULPI PHY, the USB MSIO signals are connected to four separate mutually exclusive I/O groups: USBA, USBB, USBC, and USBD I/O. In the M2S050-F896 devices, only the USBD I/O group should be used.

Excluding the M2S050-FG896, other SmartFusion2 devices' availability of the USB ULPI I/O group are package specific and some of the packages do not have any USB ULPI I/O group.
For more information, see the *DS0115: SmartFusion2 Pin Descriptions Datasheet*. In addition, USB is not supported in the VF400, FG484, and FCS325 packages for the M2S050 device.

Solution for M2S050-FG896 Devices:

Use channel D for USB I/Os with the M2S050-FG896 devices. For example, The USB_ DATA6 can be connected to the I/O through USB_DATA6_A, USB_DATA6_B, and USB_DATA6_C. Use only USB_DATA6_D in the M2S050-FG896 COM/IND device.

3.6 VPP must be set to 2.5 V when programming/writing the eNVM at Industrial temperature range

VPP can be set to 2.5 V or 3.3 V. However, when writing or programming the eNVM of the M2S050 devices below 0°C, VPP must be set to 2.5 V. For VPP minimum and maximum settings, see the *DS0128: IGLOO2 FPGA and SmartFusion2 SoC FPGA Datasheet*. Note that the eNVM reading with VPP set to 3.3 V or 2.5 V operates as intended.

3.7 Over-voltage support on MSIOs during Flash*Freeze mode

When the input voltage is driven above the reference voltage for that bank, additional current can be consumed in Flash*Freeze mode.

3.8 Verification of the FPGA fabric at junction temperatures higher than 50°C erroneously indicates a failure

Standalone verification (STAPL VERIFY action) should be run at temperatures lower than 50°C. If a VERIFY action is run at temperatures higher than 50°C, a false verify failure may be reported. Note that the Check Digest system services can be used to confirm design integrity at temperatures within the recommended operation conditions.

3.9 DDR_OUT and I/O-Reg functional Errata due to a software bug

This errata only applies if you created or updated your design using Libero SoC v11.1 SP1 or v11.1 SP2. If you have one of the following conditions in your design, the corresponding I/O will not function properly in the silicon device due to the wrong software implementation of the I/O macro.

- If you use DDR_OUT macro in your design
- If you combine an output or output enable register with an I/O using the PDC command set_io <portName> -register yes

Solution:

Both errata are fixed in Libero SoC v11.1 SP3. Migrate your design to Libero SoC v11.1 SP3 or a newer version, and re-run Compile and Layout.

3.10 Dedicated differential I/O driving the reference clock of the CCC may cause a functional failure due to a software bug

If your design has the dedicated differential I/O pair driving the reference clock of the CCC, the input clock may not propagate to CCC due to a software bug and the device will fail during silicon testing. There are several options to drive the ref clock of the CCC. One of the options is to drive from “Dedicated Input PAD x” (x = 0 to 3); this uses hardwired routing. In this option, choose single-ended I/O or differential I/O as the ref clock. This errata exists when you choose the differential I/O option, meaning the dedicated differential I/O is used as CCC reference clock input.

This errata can not be detected in any functional simulation, and can only be detected in silicon testing.

Solution:

The errata is fixed in the Libero SoC 11.1 SP3. Migrate your design to the Libero SoC 11.1 SP3 or newer version, and re-run Compile and Layout.

3.11 NVM Ready bit in eNVM Status register can generate a false READY signal

If you send an instruction to the eNVM controller and then start polling the READY signal (bit0 of the eNVM Status register) to check when the eNVM controller is ready for the next function, the first assertion of the READY signal occurs when the eNVM controller is not yet ready, resulting in the generation of a false READY signal. However, the immediate next assertion of the READY signal will correctly indicate that the eNVM controller is ready.

Workaround:

The workaround is to add an extra eNVM Status bit read that will poll/read the eNVM Status bit twice as READY.

3.12 Power-up Digest is not supported

Workaround:

The user can use NVM Data Integrity Check System service after the device is on and check the data integrity.

3.13 Programming of the eNVM should only occur as part of a bitstream also containing the FPGA fabric

The Bitstream Configuration dialog box in the Libero SoC allows the user to program eNVM and the FPGA fabric separately. However, for the current production of Smartfusion2 SoC FPGAs, the user needs to program the eNVM along with the FPGA fabric. The fabric can be programmed separately if needed.

Solution:

The errata is fixed in the Libero SoC 11.1 SP3. Migrate your design to the Libero SoC 11.1 SP3 or newer version, and re-run Compile and Layout.

3.14 Updating eNVM from the MSS or the FPGA fabric requires changes of NV_FREQRNG register

When updating the eNVM from the MSS or FPGA fabric, NV_FREQRNG register must be changed from the default value 0x07 to 0x0F, eNVM reads are not affected.

3.15 SYSCTRL_RESET_STATUS macro is not supported

3.16 Zeroization is not supported

3.17 Concurrent access of Cortex-M3 I and D-buses is not allowed

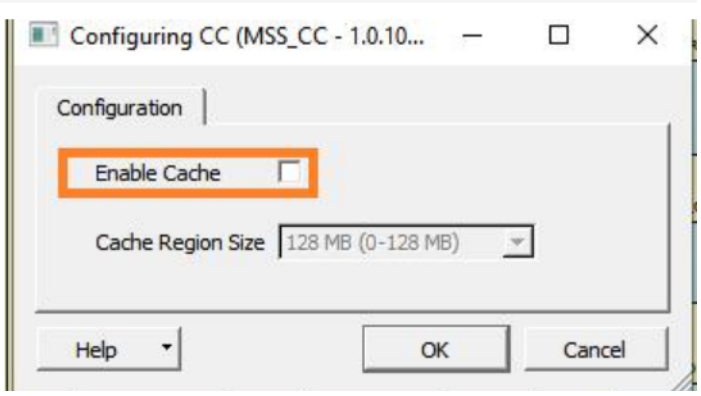
A concurrent access of the Cortex-M3 I-Bus and D-Bus may result in an invalid value returned to the internal registers from the cache when both accesses are sourced by the cache.
For more details, see the *UG0331: SmartFusion2 Microcontroller Subsystem User Guide*, Cache Controller Chapter.

The following are the possible four workarounds:

Workarounds:

- **Workaround 1:** IAR toolchain users can fix this problem by preventing the Cortex-M3 processor from issuing concurrent I and D buses access through the cache by locating the constants and data variables outside the memory regions accessed by the cache to prevent conflicts in the linker scripts. The compiler also requires adding the `-no_literal_pool` option to prevent the compiler/assembler from locating variables close to instructions known as literal pools. For more information, refer to section 4.2.3.3.1 from *UG0331: SmartFusion2 Microcontroller Subsystem User Guide*.
- **Workaround 2:** Cache can be turned off by disabling the **Enable Cache** check box in the **Cache Configurator** of the SmartFusion2 MSS Subsystem, as follows.

Figure 1 • **Disabling Cache in Cache Configurator**



- **Workaround 3:** Insert NOPs and recompile to remove the condition that causes concurrent access issues of Cortex-M3 I and D buses.
- **Workaround 4:** The application needs to be executed from eSRAM (that is, 0x20000000) instead of eNVM (that is, 0x0000000) by changing the linker script.

3.18 PCIe Hot Reset support requires a soft reset solution

On SmartFusion2 devices, a PCI Express Hot Reset requires a soft FPGA logic reset scheme which clears the sticky bits of the PCI configuration space.

Workaround:

The application note *AC437: Implementing PCIe Reset Sequence in SmartFusion2 and IGLOO2 Devices* describes the PCIe Hot Reset reset scheme. However, this reset scheme causes PCIe violations in some cases.

- For devices at Gen1 rates there are no violations.
- For devices at Gen2 rates there are two PCIe CV violations.
  - Test case 1: TD_1_7 (Advanced Error Reporting Capability)
  - Test case 2: TD_1_41 (LinkCap2Control2Status2 Reg)

3.19 After successful completion of CM3 ISP (without a SYSRESET), LSRAM Read and Write access fails from the fabric path

If LSRAM Read and Write access fails from the fabric path after performing CM3 ISP, perform a system reset or F*F Entry/Exit.

Workaround:

The user application must execute System Reset as soon as the system service is completed. Otherwise, user write and read accesses to LSRAM/uRAM will not be possible. The System Reset can be generated with the use of the tamper macro (availably in the Libero SoC Catalog). Immediately after the ISP service, the user logic checks the LSRAM/uRAM access. If access is denied, the user logic sends the reset request/interrupt to the system controller through the tamper macro (by enabling the RESET Function in the tamper macro configuration window), and then the system controller executes the system level reset.

For more information, see the *UG0451: IGLOO2 and SmartFusion2 Programming User Guide*.

The following application notes have more information and design examples on how to implement the workaround:

- *DG0471: SmartFusion2 SoC FPGA - In-System Programming Using USB OTG Controller Interface - Libero SoC Demo Guide*
- DG0454: SmartFusion2 SoC FPGA - In-System Programming Using UART Interface Demo - Libero SoC Demo Guide

3.20 Disable Cortex-M3 when programming eNVM only

The user uses the Bitstream Configuration dialog box in the Libero SoC tool and generates an eNVM only stapl file. During programming, the system controller takes control of the eNVM. If the user design has application code running from the eNVM, Cortex-M3 is halted because it cannot access eNVM. When the programming of eNVM is completed, the system controller releases eNVM.
Cortex-M3 continues running from the same address it halted at, unless the device is restarted. If the device in not restarted, the Cortex-M3 behavior will be unpredictable as eNVM has the updated code.

Workaround:

- Use the M3_Reset_N signal to hold Cortex-M3 in reset before programming eNVM
- Force a device restart
- Program eNVM and fabric

Libero v11.7 will fix the unpredictable behavior issue by forcing a restart of the device when doing eNVM only programming.

3.21 The DDR I/Os in M2S050 (T,TS)-FG896 are non-compliant with the DDR3 standard

The DDR I/Os in the M2S050-FG896 device are non-compliant with the DDR3 standard. Please contact *Microsemi Global support* for additional information.

3.22 For S (security) grade devices, user must not enable write protection for protected 4 K Regions, also known as Special Sectors in the eNVM

For S (security) devices, there are two or four 4 KB regions per eNVM array that can be protected for read and write, these regions are known as protected 4K Regions or Special Sectors. If write protection is enabled for any of these regions, none of the locked pages inside the same eNVM block can be unlocked.

3.23 User eNVM write is incorrectly allowed for the protected 4K Regions

The eNVM protected 4K Regions (0x6003FC80-0x6003FFFF) are reserved for use by the system controller to store things like the device certificate and digest information.

M2S050 rev0 and rev1 devices incorrectly allowed this eNVM area to be written. This was subsequently fixed in later versions of silicon devices to return an NVM_PAGE_LOCK_ERROR error on attempts to write this eNVM address range.

4 Usage Guidelines for SmartFusion2 Devices

Microsemi recommends the following conditions for the SmartFusion2 device usage.

4.1 Programming support

Note that there may be package dependencies that may not expose certain programming interfaces. For more information on device/package specific features, see the *DS0115: SmartFusion2 Pin Descriptions Datasheet*.

Table 3 • **Revision 0 and Revision 1 Devices**

Programming Mode	JTAG	SPI Slave	Auto Programming	Auto Update	2 Step IAP	Programming Recovery	M3 ISP
Programming Interface	JTAG	SC_SPI	SC_SPI	SPI_0	SC_SPI	SPI_0	N/A
M2S050 (T,TS)	Yes	Yes	No	No	No	No	Yes

Table 4 • **Revision 2 Device**

Programming Mode	JTAG	SPI Slave	Auto Programming	Auto Update	2 Step IAP	Programming Recovery	M3 ISP
Programming Interface	JTAG	SC_SPI	SC_SPI	SPI_0	SC_SPI	SPI_0	N/A
M2S050 (T,TS)	Yes	Yes	Yes	No	Yes	No	Yes

4.2 SHA-256 System Service

Microsemi recommends the message required to be on byte boundary when using SHA-256 System Service for the SmartFusion2 devices.

4.3 MSS reset mode

To keep the MSS in reset during normal operation, it is necessary to wait for the device to power up, and then apply the reset. The US_POR_B signal from the MSS (the power-on-reset for the FPGA fabric) can be used to check the device's powered up state.

4.4 Accessing the PCIe Bridge register in the high speed serial interface

The PCIe Bridge registers should not be accessed before the PHY is ready. Wait for the PHY_READY signal (which indicates when PHY is ready) to be asserted before updating the PCIe Bridge registers.The PHY_READY signal is normally asserted within 200 µs after the device is powered up, so wait for 200 µs before accessing the PCIe Bridge registers.